

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

- 1 1. (Currently Amended) In a communication network, a method of TCP state
2 migration comprising the steps of:
 - 3 a) establishing a TCP/IP communication session between a client computer
4 and a first server computer, said first server computer part of a plurality of server
5 computers forming a web cluster containing information, said communication session
6 established for the transfer of data contained within said information;
 - 7 b) handing off, by sending a handoff request packet, said communication
8 session to a selected server computer from said first server computer over a persistent
9 control channel using TCP handoff modules that are dynamically loadable within TCP/IP
10 stacks in operating systems located at both said first server computer and said selected
11 server computer, wherein the TCP handoff modules implement a TCP handoff protocol
12 that works within kernel levels of the operating systems, wherein said first server
13 computer includes a first bottom TCP (BTCP) module, and said selected server computer
14 includes a second BTCP module; and
 - 15 c) migrating a first TCP state of said first server computer to said selected
16 server computer, and a second TCP state of said selected server computer to said first
17 server computer over said control channel, wherein said migrating comprises migrating
18 said first TCP state to said second BTCP module at said selected server computer over
19 said control channel by including a first initial sequence number in said handoff request
20 packet, such that said second BTCP module can calculate said first TCP state for said
21 first server computer in said TCP/IP communication session.

2. (Currently Amended) The method as described in Claim 1, wherein said step a) comprises the steps of:

- receiving a SYN packet from said client computer at ~~[[a]]~~said first ~~BTCP~~ ~~bottom~~ ~~TCP (BTCP)~~-module located at said first server computer;
- sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer;
- receiving a first SYN/ACK packet from said first TCP module;
- parsing said first TCP state from said first SYN/ACK packet, including ~~[[a]]~~said first initial sequence number for said first TCP module associated with said TCP/IP communication session;
- sending said SYN/ACK packet to said client computer;
- receiving an ACK packet from said client at said first BTCP module;
- sending said ACK packet to said first TCP module;
- receiving a web request packet associated with said TCP/IP communication session at said first BTCP module at said first server computer;
- storing said SYN packet, said ACK packet and said web request packet at said first server computer.

1 3. (Currently Amended) The method as described in Claim 2, wherein said
2 step b) comprises the steps of:
3 examining content of said web request packet;
4 determining which of said plurality of server computers can best process said web
5 request packet, based on said content, wherein the server computer determined to be able
6 to best process said web request packet is the selected server computer;
7 sending ~~[[a]]the~~ handoff request packet from said first BTCP module to ~~[[a]]said~~
8 second BTCP module at said selected server computer over said control channel, if said
9 selected server computer is not said first server computer;
10 including said SYN packet and said ACK packet in said handoff request packet;
11 changing a destination IP address of said SYN packet to a second IP address of
12 said selected server computer, at said second BTCP module;
13 sending said SYN packet from said second BTCP module to a second TCP
14 module at said selected server computer;
15 receiving a second SYN/ACK packet at said second BTCP module;
16 parsing said second TCP state from said second SYN/ACK packet, including a
17 second initial sequence number, for said second TCP module, that is associated with said
18 TCP/IP communication session;
19 changing a destination IP address of said ACK packet to said second IP address,
20 at said second BTCP module;
21 updating said ACK packet to reflect said second TCP state of said selected server
22 computer in said communication session;
23 sending said ACK packet that is updated from said second BTCP module to said
24 second TCP module; and
25 sending a handoff acknowledgment message from said second BTCP module to
26 said first BTCP module.

1 4. (Cancelled)

1 5. (Currently Amended) The method as described in Claim 1[[[3]], wherein
2 step c) comprises: migrating ~~said a~~ second TCP state of said selected server computer to
3 said first BTCP module by including said second initial sequence number in ~~said a~~
4 handoff acknowledgment message, such that said first BTCP module can calculate said
5 second TCP state for said selected server computer in said TCP/IP communication
6 session.

1 6. (Previously Presented) The method as described in Claim 2, comprising
2 the further steps of:
3 intercepting a connection indication message sent from said first TCP module to
4 an application layer above said first TCP module at a first upper-TCP (UTCP) module at
5 the first server computer, said connection indication message sent by said first TCP
6 module upon establishing said communication session; and
7 holding said connection indication message at said first UTCP module, wherein
8 said first UTCP module and said first BTCP module provide a wrapper around said first
9 TCP module.

1 7. (Previously Presented) The method as described in Claim 6, wherein said
2 method comprises the further steps of:
3 sending a reset packet from said first BTCP module upon receiving said handoff
4 acknowledgment message to said first TCP module;
5 discarding said connection indication message at said first UTCP module;
6 receiving incoming data packets from said client at said first BTCP module;
7 changing destination addresses of said incoming data packets to said second IP
8 address;
9 updating sequence numbers and TCP checksum in said data packets to reflect said
10 second TCP state of said selected server computer; and
11 forwarding said updated data packets to said selected server computer.

1 8. (Previously Presented) The method as described in Claim 6, comprising
2 the further steps of:
3 sending notification from said first BTCP module to said first UTCP module to
4 release said connection indication message, if said selected server computer is said first
5 server computer;
6 sending incoming data packets, including said web request packet, from said
7 client computer, received at said first BTCP module, upstream.

1 9. (Previously Presented) The method as described in Claim 2, comprising
2 the further steps of:
3 intercepting outgoing response packets from said selected server computer at a
4 second bottom TCP (BTCP) module located at said selected server computer;
5 changing, by the second BTCP module, source addresses of said response packets
6 to a first IP address of said first server computer;
7 updating, by the second BTCP module, sequence numbers and TCP checksum in
8 said response packets to reflect said first TCP state of said first server computer; and
9 sending said updated response packets to said client computer without passing the
10 updated response packets through the first server computer.

1 10. (Previously Presented) The method as described in Claim 9, comprising
2 the further steps of:
3 monitoring TCP/IP control traffic for said communication session at said second
4 BTCP module;
5 understanding when said communication session is closed at said selected server
6 computer;
7 sending a termination message to said first server computer over said control
8 channel;
9 terminating said TCP/IP communication session at said first server computer by
10 terminating a forwarding mode at said first BTCP module; and
11 freeing data resources associated with said communication session at said first
12 server computer.

11. (Previously Presented) In a communication network, a method of TCP state migration comprising the steps of:

- a) establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;
- b) monitoring traffic associated with establishing said TCP/IP communication session to understand a first initial TCP state of said first server computer associated with said TCP/IP communication session, at a first bottom-TCP (BTCP) module at said first server computer, wherein said first BTCP module is a dynamically loadable kernel module (DLKM) loaded in said first server computer without modifying a first operating system of said first server computer;
- c) receiving a web request associated with said TCP/IP communication session at said first BTCP module at said first server computer;
- d) examining content of said web request;
- e) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content;
- f) handing off said communication session to said selected server computer from said first server computer over a persistent control channel, if said selected server computer is not said first server computer;
- g) monitoring traffic associated with handing off said TCP/IP communication session to understand a second initial TCP state of said selected server computer associated with said TCP/IP communication session, at a second BTCP module at said selected server computer, wherein said second BTCP module is a DLKM loaded in said selected server computer without modifying a second operating system of said selected server computer;
- h) migrating said first initial TCP state to said selected server computer over said control channel, such that said second BTCP module can calculate a first TCP state for said first server computer in said TCP/IP communication session;

i) sending a second initial TCP state of said selected server computer to said first BTCP module, such that said first BTCP module can calculate a second TCP state for said selected server computer in said TCP/IP communication session;

j) forwarding data packets received at said first BTCP module from said client to said selected server computer, by changing said data packets to reflect said second TCP state and a second IP address of said selected server computer;

k) sending response packets from said selected server computer directly to said client computer by changing said response packets to reflect said first TCP state and a first IP address of said first server computer; and

l) terminating said TCP/IP communication session at said first server computer when said TCP/IP communication session is closed.

12. (Previously Presented) The method as described in Claim 11, wherein said steps a) and b) comprise the steps of:

receiving a SYN packet from said client computer at said first BTCP module;

sending said SYN packet upstream to a first TCP module located above said first BTCP module in said first operating system of said first server computer;

receiving a first SYN/ACK packet from said first TCP module;

parsing, by the first BTCP module, said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with said TCP/IP communication session;

sending said SYN/ACK packet to said client computer;

receiving, by the first BTCP module, an ACK packet from said client computer at said first BTCP module;

sending said ACK packet to said first TCP module;

storing said SYN packet, said ACK packet and said web request at said first server computer.

1 13. (Previously Presented) The method as described in Claim 12, wherein said
2 steps f) and g) comprise the steps of:
3 sending a handoff request packet from said first BTCP module to said second
4 BTCP module over said control channel;
5 including said SYN packet and said ACK packet in said handoff request packet;
6 changing a destination IP address of said SYN packet to a second IP address of
7 said selected server computer, at said second BTCP module;
8 sending, by said second BTCP module, said SYN packet to a second TCP module
9 at said selected server computer;
10 receiving a second SYN/ACK packet at said second BTCP module;
11 parsing said second initial TCP state from said second SYN/ACK packet,
12 including a second initial sequence number, for said second TCP module, that is
13 associated with said TCP/IP communication session;
14 changing a destination IP address of said ACK packet to said second IP address,
15 at said second BTCP module;
16 updating said ACK packet to reflect said second TCP state of said selected server
17 computer in said communication session;
18 sending said ACK packet that is updated from said second BTCP module to said
19 second TCP module; and
20 sending a handoff acknowledgment message from said second BTCP module to
21 said first BTCP module.

1 14. (Previously Presented) The method as described in Claim 13, wherein said
2 ACK packet includes said first initial TCP state of said first server computer.

1 15. (Previously Presented) The method as described in Claim 13, wherein said
2 handoff acknowledgment message includes said second initial sequence number.

1 16. (Previously Presented) The method as described in Claim 13, comprising
2 the further steps of:

3 intercepting a connection indication message sent from said first TCP module to
4 an application layer above said first TCP module at a first upper-TCP (UTCP) module at
5 said first server computer, said connection indication message sent by said first TCP
6 module upon establishing said communication session; and

7 holding said connection indication message at said first UTCP module, wherein
8 said first UTCP module and said first BTCP module provide a wrapper around said first
9 TCP module.

1 17. (Previously Presented) The method as described in Claim 16, comprising
2 the further steps of:

3 sending a reset packet from said first BTCP module upon receiving said handoff
4 acknowledgment message to said first TCP module;

5 discarding said connection indication message at said first UTCP module;

6 receiving incoming data packets from said client at said first BTCP module;

7 changing destination addresses of said incoming data packets to said second IP
8 address;

9 updating sequence numbers and TCP checksum in said data packets to reflect said
10 second TCP state of said selected server computer; and

11 forwarding said updated data packets to said selected server computer.

1 18. (Previously Presented) The method as described in Claim 11, wherein step
2 k) comprises the steps of:
3 intercepting outgoing response packets from said selected server computer at said
4 second BTCP module;
5 changing, by said second BTCP module, source addresses of said response
6 packets to said first IP address;
7 updating, by said second BTCP module, sequence numbers and TCP checksum in
8 said response packets to reflect said first TCP state of said first server computer; and
9 sending said updated response packets to said client computer without passing the
10 updated response packets through said first server computer.

1 19. (Previously Presented) The method as described in Claim 11, wherein step
2 l) comprises the steps of:
3 monitoring TCP/IP control traffic for said communication session at said second
4 BTCP module;
5 understanding when said communication session is closed at said selected server
6 computer;
7 sending a termination message to said first server computer over said control
8 channel;
9 terminating a forwarding mode at said first BTCP module; and
10 freeing data resources associated with said communication session at said first
11 server computer.

1 20. (Previously Presented) The method as described in Claim 16, comprising
2 the further steps of:
3 sending notification from said first BTCP module to said first UTCP module to
4 release said connection indication message, if said selected server computer is said first
5 server computer; and
6 sending incoming data packets, including said web request, from said client
7 computer, received at said first BTCP module, upstream.

1 21. (Previously Presented) The method as described in Claim 11, wherein
2 each of said plurality of server computers is constructed similarly including BTCP
3 modules located downstream from respective TCP modules in the corresponding server
4 computers, and UTCP modules located upstream from the corresponding TCP modules,
5 wherein each pair of the BTCP modules and UTCP modules provides a wrapper around a
6 corresponding TCP module.

1 22. (Cancelled)

1 23. (Previously Presented) The method as described in Claim 21, wherein said
2 control channel allows for communication between all UTCP modules of the
3 corresponding server computers.

1 24. (Original) The method as described in Claim 11, wherein said plurality of
2 server computers is coupled together over a wide area network in said communication
3 network.

1 25. (Original) The method as described in Claim 11, wherein said information
2 is partitioned/partially replicated throughout each of said plurality of server computers.

1 26. (Currently Amended) A first server computer comprising:
2 at least one processor;
3 an operating system executable on the at least one processor;
4 a TCP module in the operating system;
5 a first upper TCP (UTCP) module located above the TCP module in the operating
6 system of said first server computer;
7 a first bottom TCP (BTCP) module located below said TCP module, wherein said
8 first UTCP, TCP, and first BTCP modules implement handing off a communication
9 session between the first server computer and a second node, wherein the first BTCP and
10 first UTCP modules are dynamically loadable kernel modules (DLKMs) that are
11 dynamically loadable and unloadable in said first server computer without modifying the
12 operating system,
13 wherein said communication session is established between a client computer and
14 said first server computer, said first server computer part of a plurality of server
15 computers including said second node forming a cluster that contains information, said
16 communication session established for the transfer of data contained within said
17 information;
18 wherein said first BTCP module is configured to:
19 receive a web request associated with said TCP/IP communication
20 session;
21 examine content of said web request;
22 determine which of said plurality of server computers can best process
23 said web request, based on said content, wherein the server computer determined to be
24 able to best process said web request is the second node;
25 hand off said communication session to said second node from said first
26 server computer over a persistent control channel, if said second node is not said first
27 server computer, wherein the handing off causes migration of a first TCP state of said
28 first server computer to the second node, and migration of a second TCP state of said
29 second node to the server computer.

1 27. (Cancelled).

1 28. (Currently Amended) The first server computer as described in Claim
2 26[[27]], wherein said first BTCP module is configured to further:
3 receive a SYN packet from said client computer;
4 send said SYN packet upstream to said TCP module;
5 receive a first SYN/ACK packet from said TCP module;
6 parse a first initial TCP state from said first SYN/ACK packet, including a first
7 initial sequence number for said TCP module associated with said TCP/IP
8 communication session;
9 send said SYN/ACK packet to said client computer;
10 receive an ACK packet from said client computer;
11 send said ACK packet to said TCP module.

1 29. (Previously Presented) The first server computer as described in Claim 28,
2 wherein said first BTCP module is configured to further:
3 send a handoff request packet to a second BTCP module over said control
4 channel, said second BTCP module located below a second TCP module in a second
5 operating system at said second node;
6 include said SYN packet and said ACK packet in said handoff request packet;
7 receive a handoff acknowledgment message from said second BTCP module.

1 30. (Cancelled)

1 31. (Previously Presented) The first server computer as described in Claim 29,
2 wherein said first UTCP module is configured to:
3 intercept a connection indication message sent from said TCP module to an
4 application layer above said TCP module, said connection indication message sent by
5 said TCP module upon establishing said communication session; and
6 hold said connection indication message.

1 32. (Previously Presented) The first server system as described in Claim 31,
2 wherein said first BTCP module is configured to further:
3 send a reset packet upon receiving said handoff acknowledgment message to said
4 TCP module;
5 receive incoming data packets from said client computer;
6 change destination addresses of said incoming data packets to a second IP address
7 of said second node;
8 update sequence numbers and TCP checksum in said data packets to reflect said
9 second TCP state of said second node; and
10 forward said updated data packets to said second node.

1 33. (Previously Presented) The first server computer as described in Claim 31,
2 wherein said first BTCP module is configured to further:
3 send notification to said first UTCP module to release said connection indication
4 message, if said second node is said first server computer;
5 send incoming data packets, including said web request, from said client
6 computer, received at said first BTCP module, upstream.

34. (Previously Presented) The first server computer as described in Claim 26,
wherein said first BTCP module is configured to:

- receive a handoff request from a second BTCP module located at a second server
computer over a persistent control channel, said second server computer having
established a second communication session with a second client computer, said handoff
request including a SYN packet and an ACK packet, said SYN packet and ACK packet
used for establishing said second communication session between said second client
computer and said second server computer, said ACK packet including a first initial TCP
state including a first initial TCP sequence number of said second server computer in said
communication session;
- change a destination IP address of said SYN packet to a first IP address of said
first server computer;
- send said SYN packet to said TCP module;
- receive a SYN/ACK packet;
- parse a second initial TCP state from said SYN/ACK packet, including a second
initial sequence number, for said TCP module;
- change a destination IP address of said ACK packet to said first IP address;
- update said ACK packet to reflect said second TCP state;
- send said ACK packet that is updated to said TCP module; and
- send a handoff acknowledgment message to said second BTCP module over said
control channel.

1 35.-37. (Cancelled)

1 38. (Previously Presented) The method as described in Claim 1, wherein the
2 TCP handoff modules are dynamically loadable kernel modules, the method further
3 comprising:
4 dynamically loading the TCP handoff modules in the corresponding first server
5 computer and selected server computer without modifying the operating systems of the
6 respective first server computer and selected server computer.

1 39. (Previously Presented) The method as described in claim 6, wherein the
2 first BTCP module and first UTCP module are dynamically loadable kernel modules.